Welcome to DialogClassic Web(tm)

Dialog level 05.11.05D

Last logoff: 19may06 14:22:27

Logon file405 20may06 22:52:12

*** ANNOUNCEMENTS ***

***

NEW FILES RELEASED

***Regulatory Affairs Journals (File 183)

***Index Chemicus (File 302)

***Inspec (File 202)

RESUMED UPDATING

***File 141, Reader's Guide Abstracts

***

RELOADS COMPLETED

***File 516, D&B--Dun's Market Identifiers

***File 523, D&B European Dun's Market Identifiers

***File 531, American Business Directory

*** MEDLINE has been reloaded with the 2006 MeSH (Files 154 & 155)

*** The 2005 reload of the CLAIMS files (Files 340, 341, 942)
is now available online.

***

DATABASES REMOVED

***File 196, FINDEX

***File 468, Public Opinion Online (POLL)

Chemical Structure Searching now available in Prous Science Drug Data Report (F452), Prous Science Drugs of the Future (F453), IMS R&D Focus (F445/955), Pharmaprojects (F128/928), Beilstein Facts (F390), Derwent Chemistry Resource (F355) and Index Chemicus (File 302).

***

>>>For the latest news about Dialog products, services, content<<<

>>>and events, please visit What's New from Dialog at <<<

>>>http://www.dialog.com/whatsnew/. You can find news about<<<

>>>a specific database by entering HELP NEWS <file number>.<<<

HILIGHT set on as ' '
>>>100 is not in the range between 1 and 50, original value 30 is used.
IGOR705 is set ON as an alias for
2,9,15,16,20,35,65,77,99,148,160,233,256,275,347,348,349,474,475,476,583,6-
10,613,621,624,634,636,810,813
IGORMEDIC is set ON as an alias for
5,34,42,43,73,74,129,130,149,155,442,444,455
IGORINSUR is set ON as an alias for 169,625,637
IGORBANK is set ON as an alias for 139,267,268,625,626
IGORTRANS is set ON as an alias for 6,63,80,108,637
IGORSHOPCOUPON is set ON as an alias for 47,570,635,PAPERSMJ,PAPERSEU
IGORINVEN is set ON as an alias for 6,7,8,14,34,94,434
IGORFUNDTRANS is set ON as an alias for 608
* * *
SYSTEM:HOME

*** DIALOG HOMEBASE(SM) Main Menu ***

Information:

1. Announcements (new files, reloads, etc.)
2. Database, Rates, & Command Descriptions
3. Help in Choosing Databases for Your Topic
4. Customer Services (telephone assistance, training, seminars, etc.)
5. Product Descriptions

Connections:

6. DIALOG(R) Document Delivery
7. Data Star(R)

/H = Help /L = Logoff /NOMENU = Command Mode

Enter an option number to view information or to connect to an online
service. Enter a BEGIN command plus a file number to search a database
(e.g., B1 for ERIC).
?

B IGOR705

>>> 77 does not exist

>>> 233 does not exist

>>>2 of the specified files are not available

20may06 22:52:43 User268082 Session D80.1

$0.00 0.329 DialUnits FileHomeBase

$0.00 Estimated cost FileHomeBase

$0.13 INTERNET

$0.13 Estimated cost this search

$0.13 Estimated total session cost 0.329 DialUnits

SYSTEM:OS - DIALOG OneSearch

File 2:INSPEC 1898-2006/May W2

(c) 2006 Institution of Electrical Engineers

File 9:Business & Industry(R) Jul/1994-2006/May 19

(c) 2006 The Gale Group

File 15:ABI/Inform(R) 1971-2006/May 20

(c) 2006 ProQuest Info&Learning

File 16:Gale Group PROMT(R) 1990-2006/May 22

(c) 2006 The Gale Group

File 20:Dialog Global Reporter 1997-2006/May 20

(c) 2006 Dialog

File 35:Dissertation Abs Online 1861-2006/Apr

(c) 2006 ProQuest Info&Learning

File 65:Inside Conferences 1993-2006/May 19

(c) 2006 BLDSC all rts. reserv.

File 99:Wilson Appl. Sci & Tech Abs 1983-2006/Apr

(c) 2006 The HW Wilson Co.

File 148:Gale Group Trade & Industry DB 1976-2006/May 22

(c)2006 The Gale Group

File 160:Gale Group PROMT(R) 1972-1989

(c) 1999 The Gale Group

File 256:TecInfoSource 82-2006/Jun

(c) 2006 Info.Sources Inc

File 275:Gale Group Computer DB(TM) 1983-2006/May 19

(c) 2006 The Gale Group

File 347:JAPIO Dec 1976-2005/Dec(Updated 060404)
(c) 2006 JPO & JAPIO
File 348:EUROPEAN PATENTS 1978-2006/ 200620
(c) 2006 European Patent Office
*File 348: For important information about IPCR/8 and forthcoming changes to the IC= index, see HELP NEWSIPCR.
File 349:PCT FULLTEXT 1979-2006/UB=20060518,UT=20060511
(c) 2006 WIPO/Univentio
*File 349: For important information about IPCR/8 and forthcoming changes to the IC= index, see HELP NEWSIPCR.
File 474:New York Times Abs 1969-2006/May 19
(c) 2006 The New York Times
File 475:Wall Street Journal Abs 1973-2006/May 19
(c) 2006 The New York Times
File 476:Financial Times Fulltext 1982-2006/May 21
(c) 2006 Financial Times Ltd
File 583:Gale Group Globalbase(TM) 1986-2002/Dec 13
(c) 2002 The Gale Group
*File 583: This file is no longer updating as of 12-13-2002.
File 610:Business Wire 1999-2006/May 20
(c) 2006 Business Wire.
*File 610: File 610 now contains data from 3/99 forward. Archive data (1986-2/99) is available in File 810.
File 613:PR Newswire 1999-2006/May 20
(c) 2006 PR Newswire Association Inc
*File 613: File 613 now contains data from 5/99 forward. Archive data (1987-4/99) is available in File 813.
File 621:Gale Group New Prod.Annou.(R) 1985-2006/May 22
(c) 2006 The Gale Group
File 624:McGraw-Hill Publications 1985-2006/May 19
(c) 2006 McGraw-Hill Co. Inc
*File 624: Homeland Security & Defense and 9 Platt energy journals added Please see HELP NEWS624 for more
File 634:San Jose Mercury Jun 1985-2006/May 19
(c) 2006 San Jose Mercury News
File 636:Gale Group Newsletter DB(TM) 1987-2006/May 19
(c) 2006 The Gale Group

File 810:Business Wire 1986-1999/Feb 28
(c) 1999 Business Wire
File 813:PR Newswire 1987-1999/Apr 30
(c) 1999 PR Newswire Association Inc

Set Items Description
S1 43 NOISE (20N) COUNTERMEASURE (20N) (LIST OR INFORMATION)
S2 26 S1 AND CIRCUIT AND NOISE
S3 0 S2 AND (REGISTERED (20N) COUNTERMEASURE)
S4 2 S2 AND REGISTER???
T S4/3,K/1-2
S5 24 S2 NOT S4
S6 1 S2 AND (NOISE (W) (REDUCTION OR REDUCING OR MINIMIZING))
T S6/3,K/1
S7 23 S5 NOT S6
T S7/3,K/1-23
S8 0 (CIRCUIT (20N) DESIGN????) AND (S1 NOT S2)
S9 3720 (CIRCUIT (20N) DESIGN????) AND (NOISE (S) (COUNTERMEASURE -
OR MINIMIZING OR REDUCTION OR REDUCING)) AND SELECT???
S10 7 S9 AND (SELECTING (10N) LIST)
?
T S10/3,K/1

10/3,K/1 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB

08111552 SUPPLIER NUMBER: 17349925 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Design a direct digital synthesis system.
Pikus, Gene
Electronic Design, v43, n15, p85(10)
July 24, 1995
ISSN: 0013-4872 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 7447 LINE COUNT: 00566

... both positive and negative feedback for the amplifier. At a desired frequency, the networks are designed to provide the right phase shift and

gain to cause the circuit to become unstable. Oops, I mean oscillate. With this circuit, you get to address frequency...without discontinuities in the waveform. These abilities allow you to frequency hop (ping-pong between selected frequencies) or sweep frequencies with almost any desired pattern.

A step up from the NCO...

...Good, don't vector on me now. But instead, mentally, or on paper, make a list of the possible candidates for parts, and start selecting the ones that can make your job easier. DDS chips usually have a bajillion leads... ...into a system somewhere.

When you're confident in your system design and the parts selected , order your long-lead items (ICs, sockets, connectors), and begin your detailed design. Use your...at the source.

Inductive interference is due to your circuit being magnetically coupled to the noise source, and that noise source may be your own electronics. So address this problem from both perspectives - minimizing the pickup as well as the transmission. Think back for a moment to some classes...same load. The bottom line is to do your best at comparing. Also, remember to select a part that's both fast enough to keep up with the rate coming out...

...getting that filter off the card again.

I've found that when it comes to selecting a DAC, there's usually a ton of databooks spread out all over the place... T S10/9/1

10/9/1 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB

08111552 SUPPLIER NUMBER: 17349925 (THIS IS THE FULL TEXT)
Design a direct digital synthesis system.
Pikus, Gene
Electronic Design, v43, n15, p85(10)
July 24, 1995
ISSN: 0013-4872 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 7447 LINE COUNT: 00566

ABSTRACT: The direct digital system (DDS) technology provides systems engineers with a powerful tool for analyzing complex waveforms and combining different modulation techniques, including frequency shift keying and binary phase shift keying. A DDS system is also very simple to design and requires only conventional electronic components such as logic gates and modulators. A set of guidelines for designing and building a DDS is presented.

TEXT:

From Block Diagrams Through System Test, Avoid The Problems And Pitfalls That Can Threaten DDS Design Success.

You walk into your office Monday morning after a lousy night of sleep, convinced that 2:30 A.M. is supposed to be a time for sleeping - not changing bed sheets for sick children. Then your eyes see the message, "proposal kickoff meeting - cafeteria - 8:00 A.M.". Glancing at your watch you realize you're already five minutes late. You drop your coat, grab your briefcase, and head down the hall. As you step through the cafeteria door, everyone turns their head to see their engineering department - you - walk in late.

Immediately a secretary hands you a stack of paper and heads toward the copy machine. You find a place to sit at the table, and the owner says "Jim! Glad you're here. This SOW (statement of work) explains exactly what they are looking for. You got 'til Friday to write the technical section." The conversation immediately changes to talks on contracts and marketing strategies...so you open the SOW and find the following technical description:

The system must be capable of generating an amplitude-modulated waveform consisting of 20 sine waves with unique amplitudes, summed together and then mixed with a time-varying frequency-modulated carrier. In addition, the carrier must be fittered in frequency with a time-varying amplitude-modulated linear ramp. The system must be computer-controlled, and supply quadrature output signals to drive a single-sideband modulator.

You immediately vector off and start solving the problem of summing 20 sine waves the conventional analog way: 20-op-amp oscillators fed into a single summing junction, a pc board with ground planes, ground-loop problems, bypass caps all over the place, tuning pots everywhere, guard

rings - NOT! Okay, start with square waves, filter the odd harmonics and get my sine waves, and then sum them with one large summing junction, use a pc board with ground planes, watch ground loops - DOUBLE NOT! You think, "I haven't even gotten to the section on absolute frequency and/or phase tolerances, drift specs, temperature stability...maybe 2:30 A.M. wasn't all that bad!"

The meeting ends suddenly, you snap back to reality, people file out, and your marketing guy comes up to you and says "Rumor has it that the key is a DDS system. Whatever that is."

Before throwing in your resignation papers, think of what your marketing guy said. This time he made sense. DDS stands for "direct digital synthesis." The name isn't as impressive as what it's capable of doing with today's digital technology.

Basically a DDS system will shift as much signal generation as possible into the digital realm. Once in the digital realm, all of the stuff asked for above, as well as in the section you didn't get to yet (such as absolute frequency and/or phase tolerances, drift specs, and temperature stability), becomes a very manageable task.

You may ask "How do I know?" Because we built a DDS system as special test equipment for a hardware-in-the-loop facility, to do functions similar to what was outlined above - and it worked.

A DDS SYSTEM

Let's take a closer look at what a DDS system "is and ain't" by contrasting three methods for generating a simple sine wave. Method one achieves the result completely in the analog realm [ILLUSTRATION FOR FIGURE 1 OMITTED]. Even though manufacturers have spent fortunes on stabilizing op amps, with this circuit we try to do the opposite - get the op amp to oscillate as inexpensively as possible.

The networks provide both positive and negative feedback for the amplifier. At a desired frequency, the networks are designed to provide the right phase shift and gain to cause the circuit to become unstable. Oops, I mean oscillate. With this circuit, you get to address frequency resolution, stability, temperature drift, and noise and/or interference issues all in the analog realm. Have fun!

Method two uses a phase-locked-loop synthesizer, and combines the analog and digital realms [ILLUSTRATION FOR FIGURE 2 OMITTED]. The PLL synthesizer is a closed-loop circuit, which immediately implies feedback,

loop delays, and other "opportunities."

The circuit works in the following way. The phase-detector block determines an error signal based on the difference between F.sub.in| and a fraction of the output frequency called F.sub.feedback|. That error signal is conditioned and used to drive a voltage-controlled oscillator (VCO). One interesting note is that the VCO output frequency will depend on any noise riding on this error signal as well. The VCO frequency then is divided down by the programmable counter and provides the feedback necessary to lock in the output frequency.

Simple in principle, right? That's because you aren't trying to generate complex waveforms. Try the SOW described earlier and you'll have a mess. You'll wish there was an easier way - and there is.

The third method is the DDS tech-, nique ILLUSTRATION FOR FIGURE 3 OMITTED|. Here, the sine wave is generated completely in the digital realm, in which the designer has excellent noise immunity and stability capabilities at his control. The desired frequency is entered into the numerically controlled oscillator (NCO) as a digital word, and the NCO generates a digital sampled sine-wave output that's locked to the crystal oscillator. The digital output then is converted to analog with a digital-to-analog converter (DAC).

Instead of "messing with pots and caps," you "employ crystal oscillators and digital technology" (amazing - it even sounds more exquisite). Crystal oscillators are easily obtained with 0.001% accuracies, and spurious frequency components of a DDS-generated waveform can be on the order of -90 dB - not bad for a chunk of silicon with leads hanging on it.

The verdict in contrasting the three methods: "No contest, DDS Wins". This is especially true when you need to synthesize complex waveforms utilizing and combining various modulation techniques, such as FM and AM modulation, FSK, BPSK, and QPSK.

BLOCK DIAGRAM

Now let's look at designing a DDS system. The best place to start is to TABULAR DATA FOR TABLE 1 OMITTED| completely understand the problem and requirements. Break the task up into sections and then make a simple block diagram showing all of the functions required to complete it. As you become more familiar with parts that are available and ways to apply them, you can take the block diagram one step further by drawing it out and showing the functions with major components you intend to use.

So now instead of shoving the previous SOW through a shredder, let's send it through a "reality filter" incorporating available parts and see what materializes:

An amplitude-modulated waveform can be accomplished with two digital-multiplier ICs. Two multipliers were employed because of the quadrature signal requirement, otherwise one would have worked.

20 sine waves with unique amplitudes, summed together is a simple math problem with the results stored into one UV EPROM (ultra-violet erasable programmable read-only memory device).

Mixed with a time-varying frequency-modulated carrier can be accomplished with a single IC called a numerically controlled oscillator (NCO)/modulator.

The carrier must be tittered in frequency with a time-varying amplitude-modulated linear ramp requires that the signal driving the NCO be modulated, and that can be accomplished with a multiplier/summer. A digital-ramp profile (stored in a PAL) can be multiplied by the desired amplitude profile, and then summed with the desired time-varying frequency profile - all with the same IC.

Computer controlled depends on the system requirements and the method you choose to implement the design. You may get by with simple registers to latch data, or you may have to design an interface to the computer bus.

Because everyone knows "it's an analog world," our problem isn't completed until we turn the digital signal into an analog one. The digital signal is converted with a DAC, then filtered and buffered for external use.

So, step I for designing a DDS system is the block diagram, and for our current SOW, it would contain blocks for the parts that were just identified ILLUSTRATION FOR FIGURE 4 OMITTED|. Once you have a block diagram, review it (with others, if you can) to make sure it's functionally correct, because the further you go down this DDS design path, the more expensive mistakes can be.

SYSTEM CONSIDERATIONS

After review, determine what your operating speeds have to be to obtain the desired results from each block in your diagram. For example, as a rule of thumb, you will want at least five samples per sine wave generated, and preferably ten. That would mean that the fastest sine wave loaded into the UV EPROM mentioned previously should consist of between

five and ten samples, then the slower ones are guaranteed to have at least five.

The NCO clock frequency also would be chosen, keeping in mind the need for five or more samples per sine wave. For best time- and frequency-domain results, don't try to push for less than five samples, but instead find faster parts that will allow higher clock speeds. You will be able to deal with the hardware issues of faster clocks, and the final performance will be favorable.

Now that you have your blocks and clocks in order, start searching for the digital components that will meet your functional needs and clock speeds. If possible, don't limit yourself to a logic family yet. Wait until you determine what chips are out there. Some chips may be in production becoming obsolete, come from overseas, have long lead times, or have more internal functions than others. Don't get bogged down in the search but instead give it a fair shot, because the knowledge you gain will help in the next DDS system.

AVAILABLE PARTS

Let me take a moment and describe a few parts that are available on the market now, parts that might help you in your design. The first part is a numerically controlled oscillator. Take a look at Figure 3b, which is a simple block diagram of what's under the hood of an NCO. To begin with, let's start at the end of the chip. The sine ROM contains one period of a digitized sine wave. Our example has 16 address lines into the ROM, so we would have 65,536 equally spaced samples for the one period of the sine wave.

One way to generate a sine wave would be to sequentially hit every location in the ROM. Then, if we varied how fast we hit every location, we can change the frequency of the sine wave generated. While this approach is simple and works, it has many implementation problems - especially when trying to generate high-frequency sine waves.

For example, let's say I want to generate a 10-kHz sine wave. I'd have to be able to operate the electronics with a clock speed of 65,536 x 10,000 = 655,360,000 Hz (what's wrong with this picture?). I could lower how many samples are in the ROM and solve my clocking-speed problem, but if you think about it, you'll find there are other drawbacks with this approach.

Fortunately, someone came up with a clever idea: Index the ROM at a constant rate, and instead of hitting every consecutive value in the ROM,

let's skip a few. Then, the output frequency will depend on how many ROM samples we skip. If we take a sample, skip three, then take a sample, skip three, etc....we'll end up with a faster frequency then if we take a sample, skip two, take a sample, skip two, and so on. The more samples we skip, the faster we will complete the TABULAR DATA FOR TABLE 2 OMITTED| sine-wave period.

An NCO implements this idea by using the upper bits of a phase accumulator to index the ROM. The accumulator's output is a running summation of the previous accumulator's output and the operator's digital input. The digital input is related to how many samples get skipped, as is the accumulator's output.

Stay with me now 'cause it gets better. Because we use the upper bits of the accumulator to index the ROM, and we have access to the accumulator's lower bits on the input, we can actually specify fractions of sample skipping, and the feedback for the accumulator will keep track of the fractions. For example, if I want to take a sample then skip 5.5 samples, I can. Then the process would be like this: take a sample, skip five, take a sample, skip six, take a sample, skip five, take a sample, skip six, and so on. This allows me to have fine resolution on the frequency I generate, all the way up to maximum usable limits of the NCO.

Here are two areas where the part really shines. It can generate frequency-stable sine waves with a resolution down to 0.008 Hz over its entire range (0 Hz - Nyquist cutoff). It's also capable of jumping the output frequency to a new value within one clock cycle (less than 50 ns) without discontinuities in the waveform. These abilities allow you to frequency hop (ping-pong between selected frequencies) or sweep frequencies with almost any desired pattern.

A step up from the NCO is a part called an NCO/modulator. It functions like the NCO (it generates a controllable sine wave) but also allows designers to mix/multiply the sine wave with a digital pattern. This digital pattern usually comes into the device on a separate set of pins. The final output will depend both your digital input pattern and the NCO sinewave.

For example, if your digital pattern is a "constant value," the device will output a single sine wave whose amplitude depends on the "constant value." Change the value and the amplitude will change. If you put a sine-wave pattern in, the device can multiply the NCO output with the

desired sine-wave pattern, and you will get a composite signal consisting of two sine waves. One sine wave's frequency equals the sum of the NCO and the pattern sine wave's frequency; and one sine wave frequency equals the difference between the NCO and the pattern sine wave's frequency. Huh - imagine that! Your trig classes may be applicable to your field after all.

Another device is a digital multiplier. It's functionality is no surprise - it does what its name implies! It usually has two input ports and one output, and will multiply two digital patterns together. If the patterns are constant, your output is constant. If they're dynamic, your output is dynamic. Now place this part into a system. Put a digital sine wave in one port and a slower changing pattern in the other, and you get an amplitude-modulated sine-wave output.

Let's consider one last component: a multiplier/summer. This part has four ports. Two input ports are multiplied together and then summed with a third input port. The result comes out the fourth port. That's simple enough. The trickiest part when using this component (and similar ones), though, is when you don't have signals covering the full input ranges (or matching data formats), and you want to maximize the number of significant output bits. You may have to think for a bit to optimize the part.

For example, an LMS12 has the following equation: Output = Input A * Input B + Input C; and port configurations of 12 bits each for port A and B, 25 bits for port C, and 26 bits for the output. Now, let's say you're faced with the following problem. Input A is a 7-bit 2's complement number, B an 8-bit off-set-binary number, and C a 20-bit inverted 2's complement number. The object is to obtain 12 significant bits on the output? Have fun!

Do you get an idea of what's out there? Good, don't vector on me now. But instead, mentally, or on paper, make a list of the possible candidates for parts, and start selecting the ones that can make your job easier. DDS chips usually have a bajillion leads, so remember that someone has to wire that puppy. If you can find some parts with multiple internal functions that your system requires, you can probably save on wiring costs later. Even if you go from schematic to pc board, you would have less lines to deal with.

TRADE-OFF DECISIONS

Tough trade-off decisions always spring up when pushing the technology envelope. It's even more exhilarating when the only parts available are

from a start-up company, or a new production run. If you find yourself in this situation, try to get a sample part for testing. Wire a prototype board and test the part all by itself, under your requirement specifications (temperature, vibration, clock speeds). It's a lot easier finding the weakest link when it's sitting alone than when it's nestled into a system somewhere.

When you're confident in your system design and the parts selected , order your long-lead items (ICs, sockets, connectors), and begin your detailed design. Use your block diagram to make intelligent decisions on how to section your circuitry, especially if you will need more than one board to complete it. If you have a good block diagram, your layout will be easier. Remember to keep your high-speed digital electronics together, rather than spreading it out, and physically separate the digital from any analog electronics. That way, you will have cleaner signals when you're finished.

You'll also have a better chance of ending up with a good system if you plan to use a good-quality card for placing components on. The card should have a solid ground plane that ideally would allow you to mount chip capacitors near the ICs for bypassing purposes. Chip caps have less inductance than leaded ones, and are recommended for bypassing high-speed ICs (including analog types). Remember that the idea behind the bypass capacitors is to provide a "localized" power source for the IC during signal transitions. So place them as close to the power lead as possible. Our "inhouse" guidelines suggest using one 0.1- micro|farad| ceramic bypass cap for each digital-IC power pin.

Let me address a typical question you may have to ponder. For whatever reason, I can't use a pc board, so should I wire wrap or solder the electronics? Well I've had good results going down both paths, because digital-electronic circuits are very forgiving. In almost all cases (with frequencies less than 40 MHz), you can choose either and wind up okay.

You might let the following questions help direct your path, though. Does the technician prefer to wire using wire wrap or by soldering? Will your IC package configurations (PGAs, leadless chip carriers, etc.) be better suited for one approach? Some board manufacturers have areas on the board specifically designed for PGAs. What board types are available and can meet your schedule and costs? Do you have "chip clips" for your logic analyzer, or will the pins on a wire-wrap board have to be used for the

analyzer leads instead?

DDS GOTCHAS

You're probably used to fixing standard "gotchas," such as solder balls, bent pins, and missing and misplaced wires, during build and test. But here are some "elegant gotchas" to fix during the design process. Check the input and output format of the chosen ICs. Is it 2's complement, complementary offset binary, or offset binary (Table 1)? If you mismatch the formats when connecting ICs, you will see some really strange waveforms during testing. Don't forget the formats of any sine waves sitting in PROM or RAM as well.

Speaking of sine waves in memory, make sure of the starting and ending samples so that complete sine-wave TABULAR DATA FOR TABLE 3 OMITTED| cycles are generated. If you miss a sample in the cycle, it will generate a noise spike in the output spectrum - ask me - I know. You also need to watch out so that you don't put an accidental offset in the sine waves. For example, an 8-bit number has 256 levels. If you take one level out for representing 0, that leaves 255 levels. That means you have 127 levels on one side of 0 and 128 levels on the other side. So don't normalize your amplitudes to the 128-level side - like we did.

Along with the formats, watch how you handle unused most-significant data-bit inputs. Pay attention to whether they should be tied high, low, or be sign-extended (toggling with the most-significant data bit you have). Also remember that signals can be gained by two or attenuated by two (intentionally or accidentally) simply by shifting data buses to the left or right one bit.

CLEAN UP

Plan ahead for "cleaning up" digital signals. When testing your electronics later, you will probably find that some digital signals are overshooting the maximum high logic level, under-shooting the minimum low logic level, or probably ringing at both levels. You should clean up the transitions so they stay within the logic-level limits. If you planned ahead during the design process, this task will be less traumatic. So expect to use some form of line-termination networks for fast signals traveling any significant distance, especially when they leave the board.

If you look through logic-design handbooks, you'll find numerous ways to terminate a line in an effort to smooth signal transitions. I'm only going to address three simple types that I've had the best results with. By

placing a small resistor (usually between 50 to 200 ohms|) between the output of the logic gate, and the input of your line, you can usually clean up the signal at the receiving end quite nicely ILLUSTRATION FOR FIGURE 5a OMITTED|. Place the resistor as close to the logic-gate output as possible. You could start with a 500- ohms| pot, and while monitoring the receiving end on a scope (using a 10X probe), adjust the resistance of the pot until the desired results are obtained. Then remove the pot from the circuit, measure its resistance, and place a fixed resistor in its place.

Here is what's happening by inserting the series resistor: The amplitude of the signal that actually travels down the line is a fraction of the logic gate's output. The fractional amount is determined by the resistor-divider network that was generated with the series resistor and characteristic impedance of the line. Now, once this reduced signal reaches the end of the line, the reflection that occurs because of the mismatched load adds to the decreased amplitude, and jumps the level up to nearly the final value. Without the resistor, a larger signal travels down the line, and the added reflection causes the level to overshoot. The reflective wave then travels back up the line toward the logic-gate end, and will eventually dampen out.

If you want to monitor the output of the logic gate, look at either the receiving end or at the logic-gate output pin, because the signal will be distorted between the series resistor and signal line. One final note on this series-resistor method is that it also works great with analog circuitry. For best results, you just need a low-impedance source and a high-impedance receiving end.

The second method for cleaning up signal transitions utilizes both the series resistor and a capacitor ILLUSTRATION FOR FIGURE 5b OMITTED|. Inserting the capacitor creates a single-pole low-pass filter, which will roll off the signal's bandwidth and reduce overshoot and undershoot. Obviously, you will want to size the capacitor to maintain good signal integrity and signal-transition times. This method helps reduce any additional noise on the signal as well.

The third method adds a resistor in parallel to the capacitor ILLUSTRATION FOR FIGURE 5c OMITTED|. The result is an attenuated signal and noise level due to the loading effect of the added resistor. Choose the resistor carefully to ensure that signal amplitudes meet logic-threshold requirements.

DIGITAL TIPS

Here are some closing digital tips to keep in mind during your design work that will help with the initial turn-on of the system. Watch when switching between logic families. Don't assume that all logic chips are created equal. Look at the input- and output-drive levels of interconnected ICs, and make sure they match. One case in point involves HC input-threshold levels - they are higher than what TTL output guarantees.

Count the pins on your schematic for each chip and make sure all inputs are tied somewhere, unless they have internal pull-ups. CMOS ICs especially need to be watched, because depending on the chip, a floating input can stress the internal output-drive transistors by turning both on simultaneously. One transistor screams "Go high," while the other yells "No, you fool, don't you know nothing? He wants the output low." In the end, the "heated" argument leads to a "heated, stressed-out" part.

You're going to find out (if you don't know yet) that most DDS ICs will contain multiple power and ground pins. A manufacturer doesn't put them there just to anchor the chip to the board; they are there to minimize internal noise in the IC. A good practice is to bypass each one separately. Now, if you lay out a pc board, watch so you don't treat the IC as a fuse, like one designer I know accidentally did. His "oopsicle" was that he wired power from the bus to a power pin on an IC. Then from another power pin on the IC, he took power and went to a second IC. Ta da! A high-speed, very expensive fuse.

A DDS system will require at least one oscillator, and most oscillator ICs are not designed to drive a bus. So buffer the clocks first before sending them on their merry way, especially if you're going between boards. I've had oscillators that worked fine until you try to drive more than a few loads, at which point the output went to pieces. Can you guess who buffers every oscillator now, even if it's not leaving the board? If you fan out the oscillator through multiple buffers, watch for clock skew so that IC data setup-and-hold time specifications aren't violated.

Hey, it may not look flattering, but wear a grounding strap when handling the ICs. The DDS chips are pricy enough without having to replace them because you popped 'em one with static electricity. You don't like getting shocks in the winter - they don't like it either.

COUPLED INTERFERENCE

Let's spend some time and talk about the analog section of your DDS

system. The ideal setup would have the digital ticky-talks far away from the analog circuitry. The reason is simple: To obtain spectrally pure waveforms you want analog circuitry shielded from noise, and noise is something digital circuitry is accustomed to generating. Digital noise and/or interference originates with the fast signal transitions. Faster rise times mean higher bandwidth and associated current requirements, which, in turn, leads to transition spikes, and a higher potential for creating noise.

So there's one of the many sources of noise. Now how does it get into the analog circuitry? Well, there are four main ways of coupling noise and/or interference into electronic circuits: radiative, inductive, capacitive, and conductive. Because numerous books have been written about noise, I won't write another one now, but I will briefly address each coupling mechanism and hopefully increase your ANR (awareness-to-noise ratio) during your design process.

Radiative-coupled interference is when your electronics are sitting in an electromagnetic field and begin to act like an antenna, picking up unwanted signals. To minimize or eliminate radiative noise, you can absorb it or reflect it before it gets to your electronics, or possibly seal it in at the source.

Inductive interference is due to your circuit being magnetically coupled to the noise source, and that noise source may be your own electronics. So address this problem from both perspectives - minimizing the pickup as well as the transmission. Think back for a moment to some classes you took. Remember learning that you could generate a magnetic field by sending a current through a loop of wire? Ever seen $V = L \times di/dt$ before? Well it all still works today. So let's apply these tidbits of knowledge to minimize inductive interference.

Every time you send a signal down a wire, it makes a "loop" with the return path and generates a magnetic field (usually very tiny, but the principle is still there). Therefore, minimize the mutual inductance paths in your circuit by decreasing loop areas (distance between signal lines and their return paths, as well as the signal length itself). Remember that mutual inductance drops off with separation, so separate potential noise sources from potential receivers. Decreasing di/dt will reduce the V (see earlier equation), so try to get by with the slowest possible rise times on your logic. Finally, high-permeability materials exist that can help shield

critical areas and components.

Capacitive coupling makes your electronics susceptible to a changing electric field. Remember I = C x dv/dt? We're not really talking about the capacitors you deliberately used - we're usually talking about stray capacitance linked to high-impedance sources that you have to find. So to help minimize this type of interference, make sure you try to cross signals at 90 degrees| to each other; don't have unterminated shields; don't have floating lines in cables; don't use higher resistor values then is necessary; and don't have long, unshielded, parallel conductors.

You can apply "capacitor physics" to the problem and reduce the capacitive coupling by: separating the "plates", reducing the "plate" area, or messing up the dielectric by inserting a shield. Where should you tie the shield? Well, all signals look for a return path to the source. Help your interference signals out by determining where they're trying to go, and tie your shields to the same spot.

Conductive coupling is when your electronics have a metallic (physical) connection to the noise source. Conductive coupling is best handled by watching how you run your analog and digital power and ground leads. So watch for ground loops - oopsicles - and apply the fact that electricity takes the path of least impedance, not resistance. Keep your signals close to your ground plane, and feel free to have separate digital and analog ground planes and returns.

THE DAC

Let's address some of the issues you're going to face in getting an analog signal out of your DDS system. You will need to use a DAC. In a nut shell (or should I say plastic or ceramic shell), a DAC can generate discrete analog-voltage levels with a magnitude that depends upon the digital input data. If you haven't worked with many DACs before, you will find that two main types exist - either they generate a current output or a voltage output. You'll also find that the faster DACs are typically of the current-output variety.

Now, if you need a voltage waveform and you only have a current-output DAC, here are two ways to solve the problem. First, you can change the current DAC's output to voltage simply by connecting a resistor to its output ILLUSTRATION FOR FIGURE 6 OMITTED|. While the circuit is simple, it's probably only good for monitoring purposes because any load on the signal will change the resistance value and thereby change the output

levels. Therefore, you may want to buffer the signal as well.

One caution for this circuit is to watch the "compliance" specification on the DAC. That's the maximum voltage you can have on the output pin of the DAC and still expect it to operate correctly. For instance, if your DAC can output 1 mA of current for full scale, don't hang a 1-M Omega| resistor on the output pin and expect to generate 1000 V for some digitally controllable "Jacob's Ladder." It won't work quite like that.

Second, you can generate an inverted voltage waveform if you hook the current-DAC output directly into the inverting input of an amplifier and provide a single feedback resistor ILLUSTRATION FOR FIGURE 7 OMITTED|. This circuit also will provide some drive capability.

Within the two main categories of DACs (voltage out and current out), you will find numerous features and capabilities. Let's highlight a few.

Latches: While a conversion is in process, the digital inputs to the DAC's internal current sources must remain constant. One of the best ways to accomplish this is with digital latches. You clock the data into the latch and it stays constant until the next value is latched in. No big deal. Well, almost no big deal. The world isn't perfect, and neither are latches. You're almost guaranteed that the latch output signals will have skew in them, and that no two lines will change at exactly the same time.

The skew is interpreted by the DAC as a command. So if you toggle 8 out of 10 bits, your DAC will probably try to go to seven different values before settling onto the final one. This DAC response generates an unwanted glitch in the output ILLUSTRATION FOR FIGURE 8 OMITTED|. Some DACs have internal latches, and others require external ones. If your DAC requires an external latch, pick one to minimize skew.

Internal track-and-hold amplifier: In an effort to reduce output glitches, some DACs even have internal track-and-hold (T/H) amplifiers. When the DAC is converting a new value, the T/H amp is placed in the hold mode. Once the conversion is complete, the T/H amplifier is set to track again. This usually is transparent to the user, and you reap the benefits without a hassle.

Reference source: The DAC will generate a maximum output that depends on a reference source. Here again, some DACs' references are internal, and others require external references. You can design and build external references that are more accurate and/or stable than some internal

references. You decide if it's worth the effort.

DAC SPECIFICATIONS

Well those are some of the available DAC features, now let's look at DAC specifications pertinent to your system. You should focus your attention in four main categories: resolution, accuracy, speed, and spectral purity.

The resolution of a DAC indicates how many discrete analog steps will exist over the output span. If n is the number of bits, then the number of steps is equal to 2n - 1. In theory, the more bits there are, the finer the resolution you will have. But, in practice, your noise sources will eventually obscure further resolution.

If you think that once you know your resolution, you know how accurate your DAC is, think again. If you look closely at the data sheets, you'll see accuracy specs that need to be considered in comparing DACs with equal resolutions. Picture a straight line from the endpoints of a DAC's output. The accuracy specs try to show you how far from the straight line you could be for a digital input code. Relative accuracy is a maximum deviation from the ideal. Differential nonlinearity is the worst-case error between the output and ideal for two adjacent codes (step change). The larger the errors, the more distorted your waveform will be.

The speed of a DAC is a parameter that can be difficult to use in comparisons. That's because each manufacturer tests its parts differently, and there's seldom an overlap between test conditions to do a one-to-one comparison. I'm not saying manufacturers present data to make their own parts look good, but they sure don't seem to present it to make your job easier.

You need to look at a couple of parameters to get a complete picture of your DAC's performance. The settling time indicates how long the DAC needs to get to a desired value, and then hold it within some tolerance. It says nothing about overshoot, so hopefully they have scope traces in the data sheet to give you an idea of what you're up against.

Slew rate tells you the maximum rate of change in the output that the DAC can deliver. It doesn't take a rocket scientist to know the rate will rely on the capacitive load - and manufacturers don't all use the same load. The bottom line is to do your best at comparing. Also, remember to select a part that's both fast enough to keep up with the rate coming out of your digital electronics, and can accurately generate the waveform you

need.

Under the category of spectral purity, there are three key specs to look at. First, glitch energy (besides sounding highly technical) is the impulse that occurs when your digital inputs are changing states. If you look for a DAC that has a small glitch energy spec, your waveform will look cleaner. You can try to filter the glitch energy out of a bad DAC, but the best thing to do is find a DAC that doesn't have it to begin with.

Second, harmonic distortion is a measure of how well a DAC can generate a pure sine wave without generating a bunch of harmonic stuff at the same time. If your DAC has lots of harmonic distortion and you're sweeping your DDS system in frequency, plan on seeing all kinds of junk in the frequency realm moving up and down. We saw it, why should you be any different?

Finally, the signal-to-noise ratio (SNR) indicates how much noise the DAC will put into the output. The theoretical ideal value to shoot for is SNR = (6.02n + 1.76) dB, where n is equal to the number of bits for the DAC.

If you're after spectral purity in the output waveform and can tolerate a delay, plan on putting an analog filter after the DAC. It will help smooth the stair-step transitions of the DAC (I'm not talking about the glitch energy here - solve that problem with a better DAC). Set the cutoff frequency of your filter beyond the highest frequency you're interested in generating, and make sure the final amplitude of the signal will be large enough after the filter. When you multiply two frequencies together and get the sum and difference terms, and the separation between the terms is large enough, you may be able to use the filter to eliminate one of the terms that you don't want.

If you do place a filter in your system, and it has a metal can for a shield that can be tied to ground, don't be cheap on solder and only tie one spot to ground. Tie the case to ground in a number of places. I never would have thought that mattered - ground is ground with a heavy chunk of metal, right? That's what I thought before trying to chase down the source of noise in my system. After scratching my head and much prayer, I "accidentally" touched a ground lead to the center of the filter case (even though I had the two ground pins on the filter already tied to ground), and the noise went down. Now you can forget about ever getting that filter off the card again.

I've found that when it comes to selecting a DAC, there's usually a ton of databooks spread out all over the place with markers between pages and...well you get the picture. So what I do sometimes is generate a table and fill in the parameters (Table 2). If Table 2 is something you can use, then copy it and use it.

PLAN FOR TESTING

Also, during the detailed design, try to keep in mind how you're going to test the system when it's finished. If you're an analog specialist, become familiar with a logic analyzer, because most of the DDS processing is accomplished in the digital realm. Plan on using a logic analyzer with a state-machine capability, because then you can use your system clock to load data into the analyzer, rather than sample the data with the analyzer clock. Hopefully, you choose a logic analyzer that allows the state-machine data to be viewed as a table and a waveform.

If you're a digital specialist, plan on using a wide-bandwidth analog scope to ensure quality signal transitions, as well as quality output analog-signal integrity in the time domain. In addition, a spectrum analyzer also will be required to ensure spectral purity of the synthesized waveforms in the frequency domain.

Each one of these test-equipment items connects to the electronics in its own way. You can hook up zillions of logic-analyzer leads to check out the electronics, or use a ribbon style connector mounted to the board and wired to the points of interest. With the latter approach, you can obtain an adapter connector for the analyzer that will plug directly into the ribbon connector, simplifying setup [ILLUSTRATION FOR FIG. 9 OMITTED].

Analog connections to the system have different needs. Lead inductance and capacitance (especially in the ground lead) can make a signal look sick. Sharp edges will usually overshoot and ring with sub-quality testing methods. So here are some guidelines: Place your 1X scope probes (and any home-grown RG58 coax cables with easy clips on them) in a box and save them for some other project. Step up to the plate, splurge, and get some quality 10X probes with matching probe-tip adapters.

Probe-tip adapters are little pieces of metal that are formed to slide a scope probe tip into, and they connect the probe's shield probe to ground in a direct path [ILLUSTRATION FOR FIG. 10 OMITTED]. Electrically they work great; mechanically they have "features" that can test your patience when under pressure. If you manhandle them, you can bend them, break solder

connections, or break off their tabs. You'll probably discover some other unique "features" on your own. They're still highly recommended, though, so plan on placing them in strategic locations for testing your analog circuitry.

While sockets may add inductance to signal lines and distort edges, they're adept at isolating sections of the circuit during testing and debug. In addition, they can substitute for ICs you think are bad (but usually aren't).

One last comment on testability during the design phase: Think of ways to get test patterns into the electronics. That may be as simple as loading PROM contents with walking ones and walking zeros, or single sine waves rather than multiple ones. Or maybe it's as simple as providing a pattern-generator input connector for a logic analyzer. The idea is to hasten testing, and to simplify waveforms being synthesized so you can tell if things are working "digitally" before including the analog.

CLOSING COMMENTS

A valid question to ask is, "Where does a DDS system run out of steam?" The most encompassing response is: When you approach the Nyquist frequency on any of the waveforms you are generating. Try to stay with at least the five samples per sine wave as was discussed earlier. I've used NCO parts that are spec'd at over 30 MHz, and even though I didn't max out the part, I believe it would have worked just as well at full speed.

Let me close with a summary checkoff list you can use - the next time you're in Jim's shoes (Table 3).

Gene Pikus, a principle evaluation engineer with Alliant Techsystems, received a BSEE from the University of Minnesota's Institute of Technology, Minneapolis.

SPECIAL FEATURES: illustration; table; chart
INDUSTRY CODES/NAMES: CMPT Computers and Office Automation; ELEC Electronics
DESCRIPTORS: Digital computers--Design and construction
PRODUCT/INDUSTRY NAMES: 3573110 (Digital Computers)
SIC CODES: 3571 Electronic computers
FILE SEGMENT: TI File 148 7